Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Hybrid Memory Cube Controller IP Core User Guide

Last updated for Quartus Prime Design Suite: 16.0
UG-01152
2016.05.02

101 Innovation Drive
San Jose, CA 95134
www.altera.com

Subscribe
Send Feedback

now part of Intel

Contents

About the Altera Hybrid Memory Cube Controller IP Core

1

2016.05.02

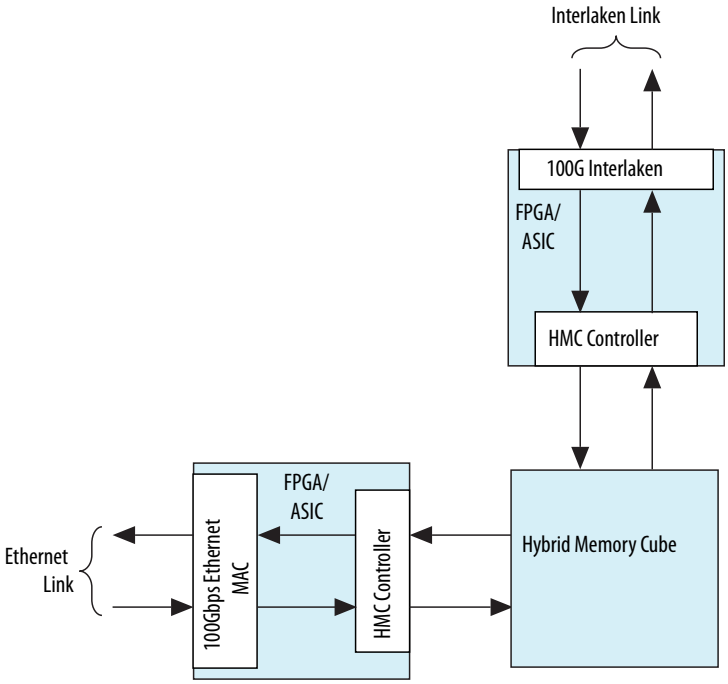
UG-01152

Subscribe Send Feedback

The Hybrid Memory Cube (HMC) specification defines a new type of memory device that provides a significant increase in bandwidth and power efficiency over existing memory architectures. The HMC specification targets high performance computers and next-generation networking equipment and provides scalability for a wide range of applications.

The Altera® HMC Controller MegaCore® IP core enables easy access to external HMC devices. HMC devices provide high bandwidth, reliable access to large amounts of memory with a small form factor, and provide significant system cost savings in high performance, memory intensive applications. The HMC Controller IP core provides a simple user interface through which you can communicate with an external HMC device to incorporate these bandwidth and performance gains in your design.

Figure 1-1: Typical HMC Controller Application



Related Information

- **HMC Controller IP Core User Guide Archives** on page 7-1

- **Hybrid Memory Cube Controller Design Example User Guide**
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.
- **HMC Specification 1.1**
  The HMC specification is available for download from the Hybrid Memory Cube Consortium web page.

HMC Controller IP Core Supported Features

The Altera HMC Controller IP core offers the following features:

- Communicates through Altera high-speed transceivers with an external HMC device compliant with the *HMC Specification 1.1*.
- Communicates with the HMC device at per-lane rates of 10 Gbps or 12.5 Gbps.
- Features Avalon® Memory-Mapped (Avalon-MM) interface to access control and status registers.
- Supports selection of a full-width variation that connects to 16 lanes of an HMC device, or a half-width variation that connects to 8 lanes of an HMC device.
  - Full-width IP core variations feature one to four simple 512-bit client data interfaces. Multiple data interfaces provide increased utilization of the HMC link.
  - Half-width IP core variations feature a single simple 256-bit client data interface.
- Supports memory READ and WRITE transactions with all valid payload sizes.
- Supports posted and non-posted versions of ATOMIC transactions, BIT WRITE transactions, and WRITE transactions.
- Supports MODE READ and MODE WRITE transactions.
- Supports optional response reordering in full-width variations, to ensure the IP core sends responses on each application response interface in the order it received the requests. When you select this option, the IP core manages the tags, which are not visible on the client interfaces.
- Supports Response Open Loop Mode for receive (RX) flow control to decrease device resource requirements.
- Supports token-based transmit (TX) flow control.
- Supports poisoned packets.
- Supports reordering of transceiver lanes for board-design flexibility.
- Supports link training sequence and provides word alignment, lane alignment, and transceiver status information in real time.
- Provides fast simulation support.
- Provides real-time error statistics.
- Provides hardware and software reset control.

- Provides power management control.
- Optionally supports ADME direct access to transceiver registers through the Altera System Console, for debugging or monitoring PHY signal integrity.
- Provides option to include ECC support in all M20K memory blocks configured in the IP core.

To support multi-link connection to the HMC device in your design, you can configure multiple HMC Controllers to communicate with the same HMC device through separate HMC links.

For the detailed HMC specification refer to the *HMC Specification 1.1*.

Related Information

HMC Specification 1.1
The HMC specification is available for download from the Hybrid Memory Cube Consortium web page.

HMC Controller IP Core Supported HMC Transaction Types

The Altera HMC Controller IP core supports all HMC transactions.

HMC Controller To HMC Device Packet Types

The HMC Controller IP core generates the following packet types on the link to the HMC device:

- NULL FLIT
- PRET (single FLIT packet)
- IRTRY (single FLIT packet)
- READ request (single FLIT packet)
- 16-byte WRITE or Posted WRITE request (2-FLIT packet)
- 32-byte WRITE or Posted WRITE request (3-FLIT packet)
- 48-byte WRITE or Posted WRITE request (4-FLIT packet)
- 64-byte WRITE or Posted WRITE request (5-FLIT packet)
- 80-byte WRITE or Posted WRITE request (6-FLIT packet)
- 96-byte WRITE or Posted WRITE request (7-FLIT packet)
- 112-byte WRITE or Posted WRITE request (8-FLIT packet)
- 128-byte WRITE or Posted WRITE request (9-FLIT packet)
- BIT WRITE or Posted BIT WRITE request (2-FLIT packet)
- MODE READ request (single FLIT packet)
- MODE WRITE request (2-FLIT packet)
- Dual 8-byte ADD IMMEDIATE or Posted Dual 8-byte ADD IMMEDIATE request (2-FLIT packet)
- Single 16-byte ADD IMMEDIATE or Posted Single 16-byte ADD IMMEDIATE request (2-FLIT packet)

The HMC Controller IP core operates in the Response Open Loop Mode and therefore does not generate TRET packets.

HMC Device to HMC Controller Packet Types

The HMC Controller IP core can process the following packet types generated by the HMC device:

- NULL FLIT
- PRET (single FLIT packet)
- TRET (single FLIT packet)
- IRTRY (single FLIT packet)
- ERROR response (single FLIT packet)
- WRITE response (single FLIT packet)
- 16-byte READ response (2-FLIT packet)
- 32-byte READ response (3-FLIT packet)
- 48-byte READ response (4-FLIT packet)
- 64-byte READ response (5-FLIT packet)
- 80-byte READ response (6-FLIT packet)
- 96-byte READ response (7-FLIT packet)
- 112-byte READ response (8-FLIT packet)
- 128-byte READ response (9-FLIT packet)
- MODE READ response (2-FLIT packet)
- MODE WRITE response (single FLIT packet)

The HMC Controller IP core does not define or support any vendor specific packet types.

Device Family Support

The following table lists the device support level definitions for Altera IP cores.

Table 1-1: Altera IP Core Device Support Levels

FPGA Device Families
Preliminary support — The core is verified with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution.
Final support — The IP core is verified with final timing models for this device family. The IP core meets all functional and timing requirements for the device family and can be used in production designs.

The following table shows the level of support offered by the HMC Controller IP core for each Altera device family.

Table 1-2: HMC Controller IP Core Device Family Support

Device Family	Support
Arria 10	Preliminary
All other device families	No support

IP Core Verification

Before releasing a version of the HMC Controller IP core, Altera runs comprehensive regression tests in the current version of the Quartus® Prime software. The HMC Controller IP core is tested in simulation and hardware to confirm functionality.

Related Information

- **Knowledge Base Errata for HMC Controller IP core**
  Exceptions to functional correctness are documented in the HMC Controller IP core errata.
- **Altera IP Release Notes**
  Changes to the HMC Controller IP core are noted in the Altera IP Release Notes starting from the Quartus II software v15.0.

Simulation

Altera performs the following tests on the HMC Controller IP core in simulation, using the Micron HMC BFM:

- Constrained random tests that cover randomized legal payload sizes and contents
- Assertion based tests to confirm proper behavior of the IP core with respect to the specification
- Extensive coverage of packet retry functionality

Constrained random techniques generate appropriate stimulus for the functional verification of the IP core. Altera monitors line, expression, and assertion coverage metrics to ensure that all important features are verified.

Hardware Testing

Altera performs hardware testing of the key functions of the HMC Controller IP core. The Altera hardware tests of the HMC Controller IP core also ensure reliable solution coverage for hardware related areas such as performance, link initialization, and reset recovery.

Altera performs hardware testing on the Arria 10 GX FPGA Development Kit with an HMC daughter card. A Micron HMC 15G-SR device on the daughter card is connected to the development board through FMC connectors.

Performance and Resource Utilization

Table 1-3: HMC Controller IP Core FPGA Resource Utilization

Typical resource utilization for an HMC Controller IP core configured with a data rate of 10 Gbps, using the Quartus Prime software v16.0, with the following IP core features turned off:

- ADME support
- M20K ECC support

The numbers of ALMs and logic registers are rounded up to the nearest 100. The numbers of ALMs, before rounding, are the **ALMs needed** numbers from the Quartus Fitter Report.

IP Core Variation			Resource Utilization		
Link Width	Response Reordering	Number of Ports	ALMs Needed	Dedicated Logic Registers	M20K Blocks
Full-width	Off	1	24400	48200	51
		2	29200	58400	87
		3	34100	68600	123
		4	38900	78800	158
	On	1	29900	59400	55
		2	37000	76200	93
		3	44200	93100	132
		4	51300	109900	170
Half-width			13400	24000	37

Related Information

- **Fitter Resources Reports in the Quartus Prime Help**
  Information about Quartus Prime resource utilization reporting, including **ALMs needed**.
- **Quartus Prime Standard Edition Handbook, Volume 1: Design and Synthesis**

Device Speed Grade Support

Table 1-4: Minimum Recommended Device Family Speed Grades

Altera recommends that you configure the HMC Controller IP core only in the device speed grades listed in the table, or any faster (lower numbered) device speed grades that are available.

Altera does not support configuration of this IP core in slower (higher numbered) device speed grades.

Device Family	IP Core Variation: Lane Rate	
	10 Gbps	12.5 Gbps
Arria 10	E1, I1, E2, I2	E1, I1

Release Information

Table 1-5: HMC Controller IP Core Current Release Information

Item	Value
Version	16.0
Release Date	May 2016
Ordering Code	Full-width: IP-HMCSR15FW Half-width: IP-HMCSR15HW
Vendor ID	6AF7

Getting Started with the HMC Controller IP Core 2

2016.05.02

UG-01152

The following information explains how to install, parameterize, and simulate the Altera Hybrid Memory Cube Controller IP core.

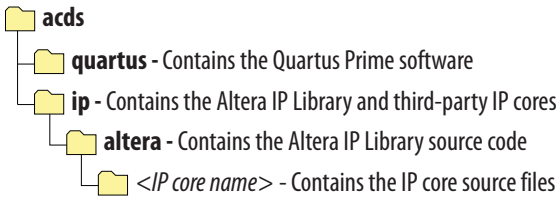
Related Information

- **HMC Controller IP Core Design Example** on page 6-1
  The HMC Controller design example provides an example of how to connect your IP core with an external I^2C master module and an external TX PLL.
- **Introduction to Altera IP Cores**
  Provides general information about all Altera IP cores, including parameterizing, generating, upgrading, and simulating IP.
- **Creating Version-Independent IP and Qsys Simulation Scripts**
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- **Project Management Best Practices**
  Guidelines for efficient management and portability of your project and IP files.

Licensing IP Cores

The Altera IP Library provides many useful IP core functions for your production use without purchasing an additional license. Some Altera MegaCore IP functions require that you purchase a separate license for production use. However, the OpenCore® feature allows evaluation of any Altera IP core in simulation and compilation in the Quartus Prime software. After you are satisfied with functionality and performance, visit the Self Service Licensing Center to obtain a license number for any Altera product.

Figure 2-1: IP Core Installation Path



Note: The default IP installation directory on Windows is **<*drive*>:\altera**<*version number*>; on Linux the IP installation directory is <*home directory*>**/altera/ <*version number*>**.

OpenCore Plus IP Evaluation

Altera's free OpenCore Plus feature allows you to evaluate licensed MegaCore IP cores in simulation and hardware before purchase. You only need to purchase a license for MegaCore IP cores if you decide to take your design to production. OpenCore Plus supports the following evaluations:

- Simulate the behavior of a licensed IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

OpenCore Plus evaluation supports the following two operation modes:

- Untethered—run the design containing the licensed IP for a limited time.
- Tethered—run the design containing the licensed IP for a longer time or indefinitely. This requires a connection between your board and the host computer.

Note: All IP cores that use OpenCore Plus time out simultaneously when any IP core in the design times out.

Related Information

- **Altera Licensing Site**
- **Altera Software Installation and Licensing Manual**

Specifying IP Core Parameters and Options

The HMC Controller parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Quartus Prime software.

1. In the IP Catalog (**Tools** > **IP Catalog**), under **Memory Interfaces and Controllers**, locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named *<your_ip>***.qsys**. Click **OK**.
3. Specify the parameters and options for your IP variation in the parameter editor. Refer to the Parameters section for information about specific IP core parameters.
4. Click **Generate HDL**, the **Generation** dialog box appears.
5. To generate a simulation model of the HMC Controller IP core, under **Simulation** > **Create Simulation Model**, select **Verilog** HDL.
6. Specify other output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
7. Click **Finish**. The parameter editor adds the top-level **.qsys** file to the current project automatically. If you are prompted to manually add the **.qsys** file to the project, click **Project** > **Add/Remove Files in Project** to add the file.
8. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

HMC Controller IP Core Parameters

The HMC Controller parameter editor provides the parameters you can set to configure the HMC Controller IP core and simulation testbenches.

The HMC Controller parameter editor includes an **Example Design** tab. For information about that tab, refer to the **Hybrid Memory Controller Design Example User Guide**.

Table 2-1: HMC Controller IP Core Parameters

Parameters for customizing the HMC Controller IP core in the **IP** tab of the HMC Controller parameter editor.

Parameter	Type	Range	Default Setting	Parameter Description
Lanes	Integer	• 8 • 16	16	Selects half-width (8 lanes) or full-width (16 lanes) functionality.
Data rate	String	• 10 Gbps • 12.5 Gbps	10 Gbps	Selects the data rate on each lane.

Parameter	Type	Range	Default Setting	Parameter Description
CDR reference clock	String	• 125 MHz • 156.25 MHz	125 MHz	Selects the frequency of the input reference clock for the RX CDR PLL. You must drive the `rx_cdr_refclk0` input signal at the frequency you specify for this parameter. In addition, your design must derive this clock, the external transceiver TX PLL reference clock, and the `REFCLKP` and `REFCLKN` input signals of the external HMC device from the same clock source.
Ports	Integer	• 1 • 2 (available only for full-width variations) • 3 (available only for full-width variations) • 4 (available only for full-width variations)	1	Number of ports (data path interfaces). This parameter is useful only for full-width variations. Half-width variations have a single port. Increasing the number of ports increases utilization of the Hybrid Memory Cube, increasing efficiency. If you specify more than one port, each port is assigned a range of tags. • If you specify 2 ports, port 0 must use tags in the range 0 to 255, and port 1 must use tags in the range 256 to 511. • If you specify 3 ports, port 0 must use tags in the range 0 to 175, port 1 must use tags in the range 176 to 351, and port 2 must use tags in the range 352 to 511. • If you specify 4 ports, port 0 must use tags in the range 0 to 127, port 1 must use tags in the range 128 to 255, port 2 must use tags in the range 256 to 383, and port 3 must use tags in the range 384 to 511.

Parameter	Type	Range	Default Setting	Parameter Description
Response re-ordering	Boolean	• True • False	False	Specifies whether the IP core ensures that responses appear on each data response interface in the order the original requests arrived on the corresponding request interface. If you turn on this feature, the IP core manages tags internally. In that case tags are not available on the data interfaces. Turning on this feature can increase round-trip latency. This parameter is available only for full-width variations.
RX mapping	64-bit value		0xFEDCBA9876543210	Selects the RX lane mapping. Use caution in modifying this parameter. Refer to **RX Mapping and TX Mapping Parameters** on page 2-7.
TX mapping	64-bit value		0xFEDCBA9876543210	Selects the TX lane mapping. Use caution in modifying this parameter. Refer to **RX Mapping and TX Mapping Parameters** on page 2-7.

Parameter	Type	Range	Default Setting	Parameter Description
Enable ADME and Optional Reconfiguration Logic	Boolean	• True • False	False	Specifies whether the IP core turns on the ADME feature in the embedded Arria 10 Native PHY IP core that configures the transceivers. Turning on this parameter turns on the following Arria 10 PHY features: • **Enable Altera Debug Master Endpoint (ADME)** • **Enable capability registers** • **Enable control and status registers** • **Enable PRBS soft accumulators** **Note:** The **Share reconfiguration interface** PHY parameter is always turned on for this IP core. The ADME feature enables Native PHY register programming with the Altera System Console, and optional reconfiguration logic. For more information, refer to the *Arria 10 Transceiver PHY User Guide*.
Enable M20K ECC support	Boolean	• True • False	False	Specifies whether the IP core supports the ECC feature in the Arria 10 M20K memory blocks that are configured as part of the IP core. You can turn on this parameter to enhance data reliability by enabling single-error correction, double-adjacent-error correction, and triple-adjacent-error detection ECC functionality in the M20K memory blocks configured in your IP core. Turn off this parameter to decrease latency and resource utilization.

Related Information

- **Arria 10 Transceiver PHY User Guide**
  Provides information about the Arria 10 ADME feature.
- **Embedded Memory Blocks in Arria 10 Devices**
  Provides information about the Arria 10 M20K block ECC feature.

RX Mapping and TX Mapping Parameters

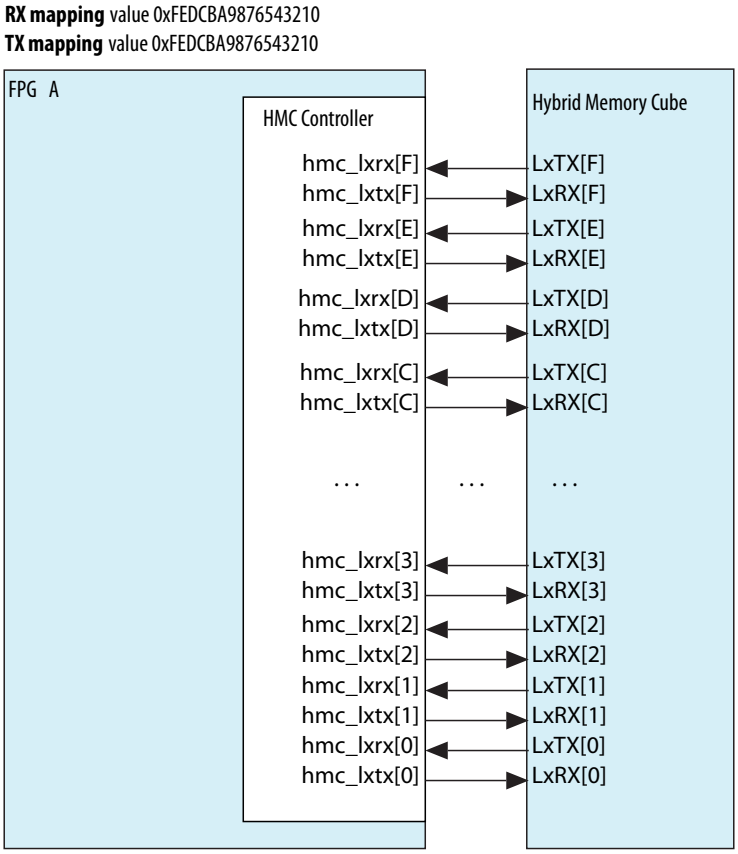
The HMC Controller IP core provides the **RX mapping** and **TX mapping** parameters for flexibility in board design.

The default values of these parameters specify the correct IP core behavior when the HMC device `LxTX[<i>]` output signal connects to the HMC Controller IP core `hmc_lxrx[<i>]` input port, and the `LxRX[<i>]` input signal connects to the HMC Controller IP core `hmc_lxtx[<i>]` output port, for each `<i>`.

However, if your design constraints prevent you from connecting these signals as expected, you can instead modify one or both HMC Controller IP core mapping parameters to accommodate the non-standard connection.

Note: The Quartus Prime Fitter prevents you from mapping the HMC Controller IP core lanes to Arria 10 device transceiver channels out of order. Therefore, these two parameters only compensate for out-of-order connections on the board between the Arria 10 transceiver pins and the HMC device ports.

Figure 2-2: Default RX and TX Mapping Parameter Values



If the HMC device LxTX[<i>] output signal connects to the HMC Controller IP core hmc_lxrx[<k>] input port, you must set the value in bits [(4<i>+3):(4<i>)] (nibble <i>) of the **RX mapping** parameter to 4'h<k>. Therefore, the default value of the **RX mapping** parameter is 0xFEDCBA9876543210, indicating that LxTX[F] connects to hmc_lxrx[F], LxTX[E] connects to hmc_lxrx[E], and so on.

If the HMC device LxRX[<i>] input signal connects to the HMC Controller IP core hmc_lxtx[<k>] input port, you must set the value in bits [(4<i>+3):(4<i>)] (nibble <i>) of the **TX mapping** parameter to 4'h<k>. Therefore, the default value of the **TX mapping** parameter is 0xFEDCBA9876543210, indicating that LxRX[F] connects to hmc_lxtx[F], LxRX[E] connects to hmc_lxtx[E], and so on.

Example: Non-Default RX Mapping Parameter Value

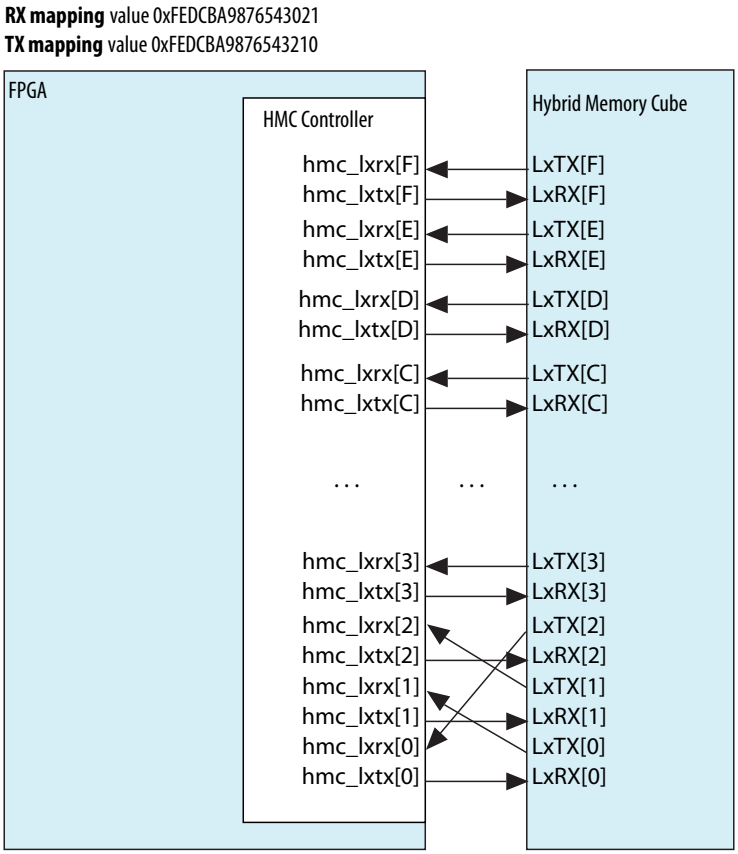
Table 2-2: Non-Default RX Connections

HMC Device Output Signal	IP Core Input Signal
LxTX[2]	hmc_lxrx[0]
LxTX[1]	hmc_lxrx[2]
LxTX[0]	hmc_lxrx[1]

Figure 2-3: Non-Default RX Mapping Parameter Value Example

If you connect the IP core `hmc_lxrx[2:0]` input signals according to the table, and connect all other IP core `hmc_lxrx[<i>]` input ports to the corresponding HMC device `LxTX[<i>]` output ports, you would set the value of the **RX mapping** parameter to 0xFEDCBA9876543**021** to compensate for the non-standard connection.

Note: The **RX mapping** parameter specifies the HMC device lane by position and the IP core lane by value. The figure illustrates a mapping parameter value of 0xFED.....43021 and not a value of 0xFED....43102.



Example: Non-Default TX Mapping Parameter Value

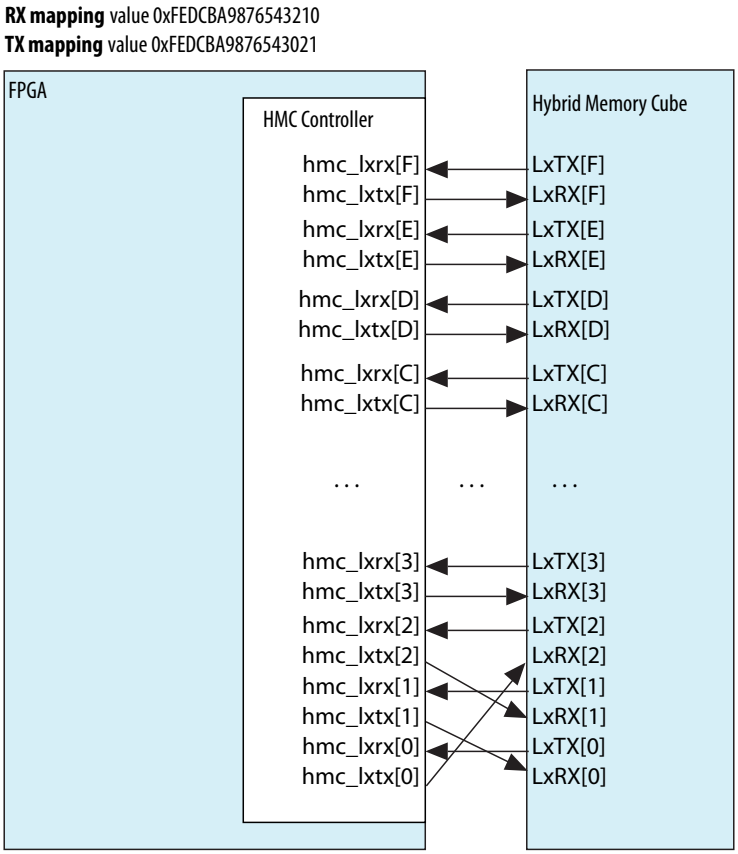
Table 2-3: Non-Default TX Connections

HMC Device Input Signal	IP Core Output Signal
`LxRX[2]`	`hmc_lxtx[0]`
`LxRX[1]`	`hmc_lxtx[2]`
`LxRX[0]`	`hmc_lxtx[1]`

Figure 2-4: Non-Default TX Mapping Parameter Value Example

If you connect the HMC Controller IP core `hmc_lxtx[2:0]` output signals according to the table, and connect all other IP core `hmc_lxtx[<i>]` output ports to the corresponding HMC device `LxRX[<i>]` input ports, you would set the value of the **TX mapping** parameter to 0xFEDCBA9876543**021** to compensate for the non-standard connection.

Note: The **TX mapping** parameter specifies the HMC device lane by position and the IP core lane by value. The figure illustrates a mapping parameter value of 0xFED.....43021 and not a value of 0xFED....43102.

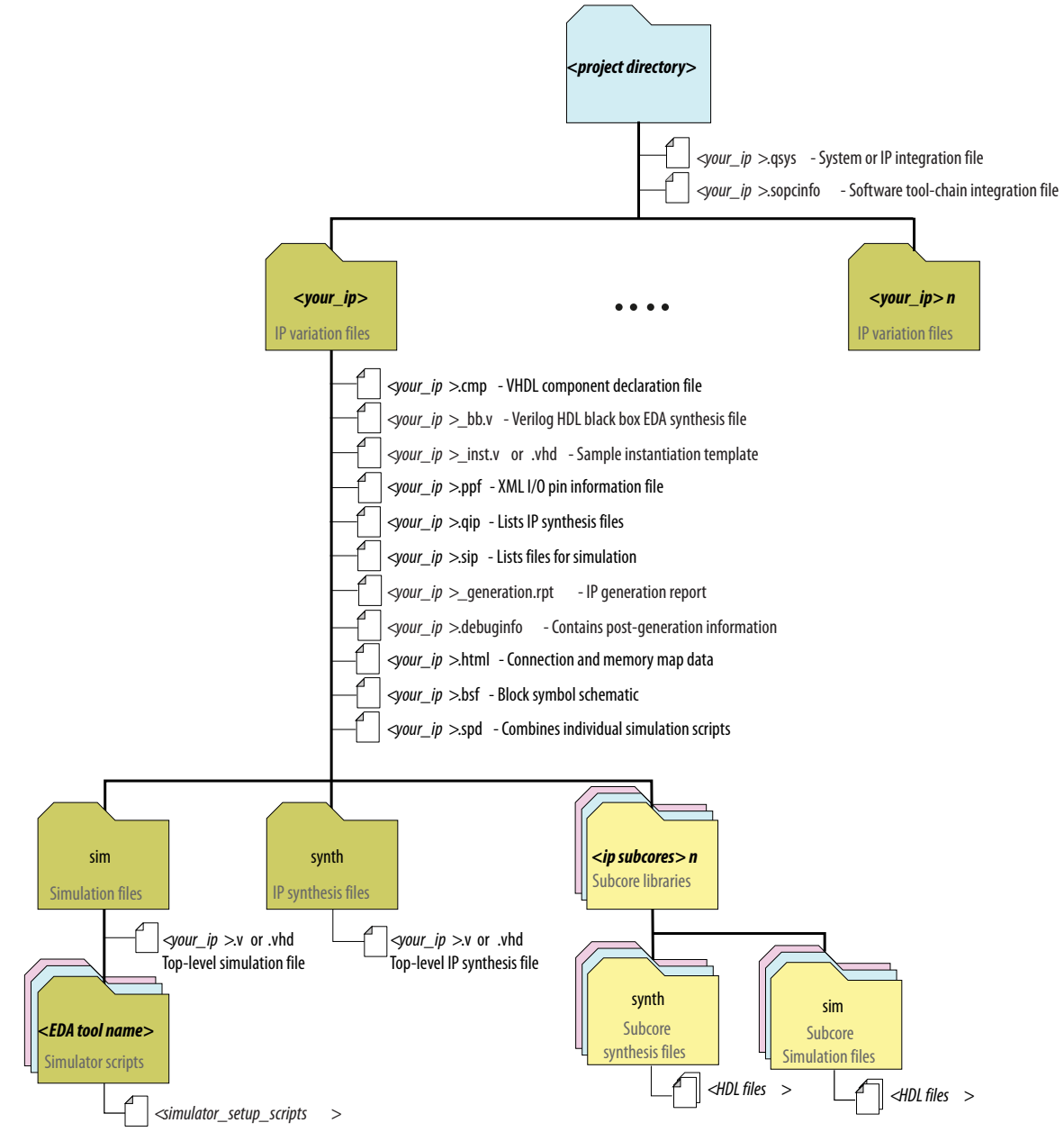


Use caution in modifying these parameters. In loopback configurations, you must ensure the **RX mapping** and **TX mapping** parameters specify reversed mappings. Otherwise, the IP core downstream of the RX lane swapper appears to receive data on the wrong lanes.

Files Generated for Altera IP Cores

The Quartus Prime software generates multiple files during generation of your IP core variation.

Figure 2-5: IP Core Generated Files



Integrating Your IP Core in Your Design

To ensure the HMC Controller IP core functions correctly in hardware, you must connect additional blocks to your IP core and assign device pins in order.

Pin Constraints

When you integrate your HMC Controller IP core instance in your design, you must make appropriate pin assignments. You can create a virtual pin to avoid making specific pin assignments for top-level signals while you are simulating and not ready to map the design to hardware.

When you are ready to map the design to hardware, you must enforce the following constraints:

- Adjacent HMC Controller lanes must map to adjacent Altera device pins. You cannot swap the lane order by mapping lanes to other Altera device pins. Instead, use the **RX mapping** and **TX mapping** parameters to compensate for board design issues.
- The lanes of an HMC Controller IP core must be configured in no more than three transceiver blocks. To enforce this constraint, you must configure IP core lanes in transceiver channels with the following restrictions:
  - Lane 0 of a full-width HMC Controller IP core must map to channel 0, 1, or 2 of a transceiver block.
    - If Lane 0 maps to channel 0, then HMC Controller Lane 1 must map to channel 1 of the same transceiver block (transceiver block N), and Lane 15 maps to channel 3 of the transceiver block N+2.
    - If Lane 0 maps to channel 1, then HMC Controller Lane 1 must map to channel 2 of the same transceiver block (transceiver block N), and Lane 15 maps to channel 4 of the transceiver block N+2.
    - If Lane 0 maps to channel 2, then HMC Controller Lane 1 must map to channel 3 of the same transceiver block (transceiver block N), and Lane 15 maps to channel 5 of the transceiver block N+2.
  - Lane 0 of a half-width HMC Controller IP core can map to any channel. If it maps to any of channels 0, 1, 2, 3, or 4, the IP core lanes are configured in two transceiver blocks.

Required External Blocks

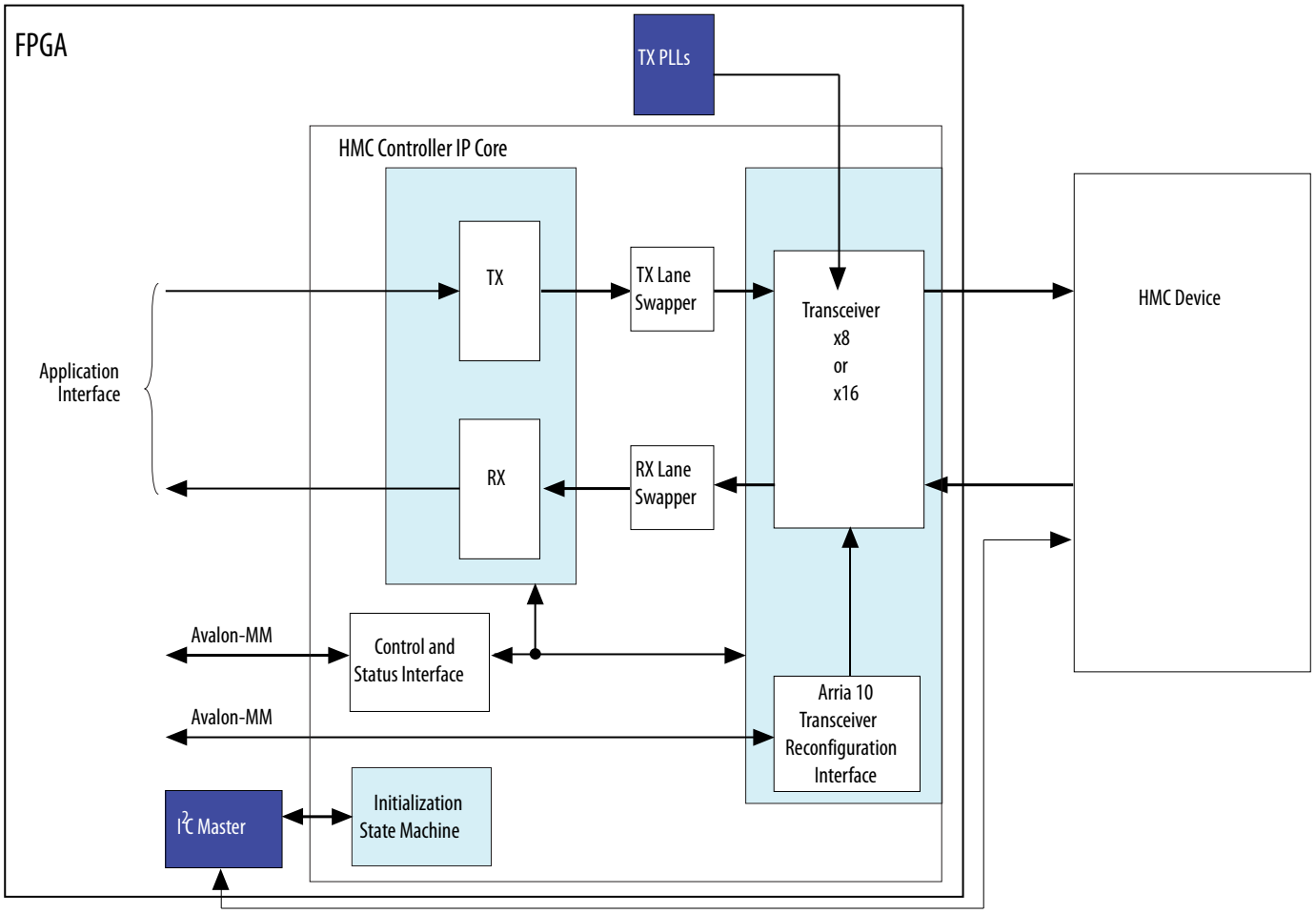
To ensure the HMC Controller IP core functions correctly in hardware, you must connect additional blocks to your IP core.

The HMC Controller IP core requires that you define and instantiate the following additional modules:

- External PLL IP core to configure transceiver TX PLL for all of the HMC lanes. Although the hardware these IP cores configure might physically be part of the device transceiver, you must instantiate them in software separately from the HMC Controller IP core. This requirement supports the configuration of multiple Altera IP cores using the same transceiver block in the device.
- An external I^2C master module in your design. Your design must include this module to initialize the HMC device to which your IP core connects.

Figure 2-6: Required External Blocks

The required external blocks appear darker than the other blocks in the figure. The external TX PLL IP core configures an ATX PLL in the device transceiver or an fPLL in Transceiver mode.



Adding the External PLL

The HMC Controller IP core requires that you generate and connect an external transceiver PLL IP core. You must generate the PLL IP core required to clock the transceiver channels that are configured as HMC Controller IP core lanes. The ATX PLL IP core configures the transceiver PLL in the transceiver in hardware, but you must generate the transceiver PLL IP core separately from the HMC Controller IP core in software. You can also configure an fPLL in transceiver mode. If you do not generate and connect the transceiver PLL IP core, the HMC Controller IP core does not function correctly in hardware.

You can use the IP Catalog to generate the external PLL IP core that configures a transceiver PLL on the device. In the IP Catalog, select **Arria 10 Transceiver ATX PLL** or **Arria 10 fPLL**.

In the transceiver PLL parameter editor, you must follow the instructions in the *Arria 10 Transceiver PHY User Guide* to configure the PLL IP core in the xN bonding configuration. In addition, you must set the following parameter values:

- **PLL output frequency** to one half of the per-lane data rate of the IP core variation. The transceiver performs dual edge clocking, using both the rising and falling edges of the input clock from the PLL. Therefore, this PLL output frequency setting drives the transceiver with the correct clock for the lanes that connect to the HMC device.
- **PMA interface width** to **32**.
- **PLL integer reference clock frequency** (ATX PLL) or **Desired reference clock frequency** (fPLL).

  Note: The HMC Controller IP core does not support PLL feedback compensation bonding.

  Altera recommends that you specify 125 MHz, 156.25 MHz, or 166.67 MHz. You can theoretically specify any reference clock frequency from which the PLL can generate the required output clock frequency. However, you must drive this TX PLL and the RX CDR PLL (`rx_cdr_refclk0` input signal to the HMC Controller IP core) and the HMC device reference clock input signals (`REFCLKP` and `REFCLKN`) from the same clock source.

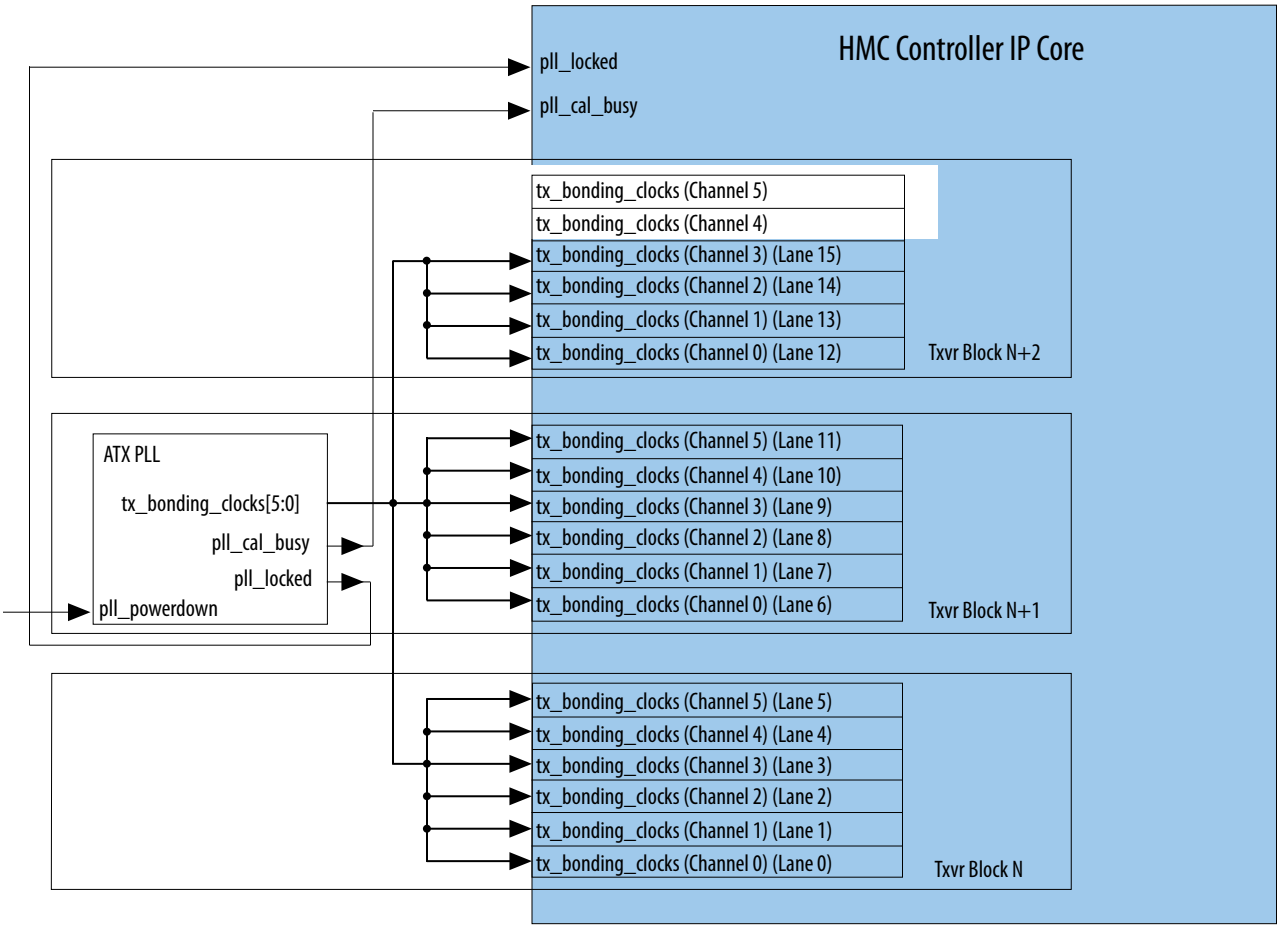
  Note: You must drive the external PLL reference clock input signal at the frequency you specify for this parameter.

In xN bonding mode, a single PLL is sufficient to drive the channels in the configured transceiver blocks. Recall that your HMC link TX serial lanes must be configured in order in adjacent physical transceiver channels so that these lanes configure a maximum of three transceiver blocks. You can view I/O constraints that enforce these requirements in the design example Quartus Settings File **hmcc_example.qsf** provided with the HMC Controller IP core.

The PLL output connects directly to the x6 network for its transceiver block and drives additional transceiver blocks through the xN clock network.

Figure 2-7: Transceiver PLL Connections Example with xN Bonding Scheme

Example connections between a full-width HMC Controller IP core and a single ATX PLL IP core in xN bonding mode.



You must connect the external PLL signals and the HMC Controller IP core transceiver TX PLL interface signals according to the following rules:

HMC Controller Signal	Connects to TX PLL Signal
`tx_bonding_clocks[5:0]` input signal for HMC lane N	`tx_bonding_clocks[5:0]` output vector of PLL IP core for the transceiver block in which lane N is configured. In the case of xN bonding, a single PLL connects to the xN clock network and the `tx_bonding_clocks[5:0]` input pins for HMC lanes in a different transceiver block from the configured PLL receive the clock from the xN clock network.
`pll_locked` input signal	`pll_locked` output signal of the external PLL for all of the HMC lanes.
`pll_cal_busy` input signal	`pll_cal_busy` output signal of the external PLL for all of the HMC lanes.